

What is Claimed is:

1. A visual programming system, comprising:

one or more function modules, each of which is provided with an applicable functional program or command stored in a computer executable language in a processing unit to accomplish a substantial application function; and

one or more programming flow lines connecting said function modules with each other in a predetermined sequence to construct a visual graphic program which is compiled to machine readable codes of said computer executable language so as to construct said visual graphic program in said computer executable language in said processing unit for executing said applicable functional programs or commands of said function modules one after another in said predetermined sequence.

2. The system, as recited in claim 1, further comprising one or more determination modules each of which is provided with a determining test stored in computer executable language in the processing unit, wherein each of said determination modules is provided with an determination entrance, a "True" exit and a "False" exit to connect with three of said function modules and other said determination modules by three of said programming flow lines to construct said visual graphic program, wherein after said visual graphic program is compiled to machine readable codes, each of said determination modules executes said determining test according to said predetermined sequence.

3. The system, as recited in claim 1, wherein each of said function modules has at least an connecting entrance and an connecting exit to connect with two of other said function modules by two of said programming flow lines, wherein each of said programming flow lines is connected from said connecting exit of one of said function modules to said connecting entrance of another of said function modules.

4. The system, as recited in claim 2, wherein each of said function modules has at least an connecting entrance and an connecting exit to connect with two of other said function modules and said determination modules by two of said programming flow lines, wherein by means of said programming flow lines, said connecting exit of one of

said function modules is capable of connecting with said connecting entrance of another of said function modules.

5 5. The system, as recited in claim 2, wherein each of said function modules has at least an connecting entrance and an connecting exit to connect with two of other
10 said function modules and said determination modules by two of said programming flow lines, wherein by means of said programming flow lines, said connecting entrance of one of said function modules is capable of connecting to one of said “True” exit and said “False” exit of another said determination modules and said connecting exit of one of said function modules is capable of connecting with said determination entrance of one of
10 said determination modules.

15 6. The system, as recited in claim 5, wherein a construction of said function modules, said determination modules and said programming flow lines is displayed by said processing unit via a monitor thereof as said visual graphic program which directly represents said computer executable language to be stored in said processing unit to
15 operate and function.

 7. The system, as recited in claim 6, wherein human readable source codes of a source code program are converted and arranged into said different function modules and said determination modules of said visual graphic program according to a conversion rules database.

20 8. The system, as recited in claim 7, further including a user editing interface to construct said visual graphic program by selecting said function modules and said determination modules and linking said function modules and said determination
25 modules being selected by said programming flow lines, and a compiler which is used to convert said human readable source code program into said machine readable codes of said computer executable language following predetermined conversion instructions of a conversion rules database.

30 9. The system, as recited in claim 1, wherein said user editing interface comprises a function module selection panel, a selected module panel, and a editorial management panel, wherein said function module selection panel comprises selectable
30 commands, including determining test commands in human readable programming languages, programming flow lines representing direction of flow of said program and

functional commands, wherein when a command is selected from said function module selection panel, said command being selected appears in said selected module panel and, by arranging said commands being selected into a flow chart form, said visual graphic program is completed.

5 10. The system, as recited in claim 8, wherein said user editing interface comprises a function module selection panel, a selected module panel, and an editorial management panel, wherein said function module selection panel comprises selectable commands, including determining test commands in human readable programming languages, programming flow lines representing direction of flow of said program and
10 functional commands, wherein when a command is selected from said function module selection panel, said command being selected appears in said selected module panel and, by arranging said commands being selected into a flow chart form, said visual graphic program is completed.

11. A visual programming method, comprising:

15 (a) assigning one or more function modules each of which is provided with an applicable functional program or command stored in a computer executable language in a processing unit to accomplish a substantial application function; and

 (b) connecting said function modules in a predetermined sequence with one or more programming flow lines, each pointing from one direction to another to construct a
20 visual graphic program; and

 (c) compiling said visual graphic program to machine readable codes of said computer executable language, wherein a logical execution sequence of said visual graph program is designated and illustrated through connection of said function modules with said programming flow lines so as to construct a finish program in said computer
25 executable language in said processing unit.

12. The method, as recited in claim 11, after the step (a), further comprising a step of assigning one or more determination modules each of which is provided with a determining test stored in said computer executable language in said processing unit, wherein each of said determination modules is connected with three of said function
30 modules and other said determination modules at an determination entrance, a "True"

exit and a “False” exit thereof by three of said programming flow lines to construct said visual graphic program, wherein after said visual graphic program is compiled to said machine readable codes, each of said determination modules executes said determining test.

5 13. The method, as recited in claim 11, wherein each of said function modules has at least an connecting entrance and an connecting exit to connect with two of other said function modules by two of said programming flow lines, wherein each of said programming flow lines is connected from said connecting exit of one of said function modules to said connecting entrance of another of said function modules.

10 14. The method, as recited in claim 12, wherein each of said function modules has at least an connecting entrance and an connecting exit to connect with two of other said function modules by two of said programming flow lines, wherein each of said programming flow lines is connected from said connecting exit of one of said function modules to said connecting entrance of another of said function modules.

15 15. The method, as recited in claim 12, wherein each of said function modules has at least an connecting entrance and an connecting exit to connect with two of other said function modules and said determination modules by two of said programming flow lines, wherein by means of said programming flow lines, said connecting entrance of one of said function modules is capable of connecting to one of said “True” exit and said
20 “False” exit of another said determination modules and said connecting exit of one of said function modules is capable of connecting with said determination entrance of one of said determination modules.

 16. The method, as recited in claim 12, wherein human readable source codes of a source code program are converted and arranged into different said function modules
25 and said determination modules of said visual graphic program according to a conversion rules database.

 17. The method, as recited in claim 16, wherein said visual graphic program is constructed by a user editing interface by selecting said function modules and said determination modules and linking said function modules and said determination
30 modules being selected by said programming flow lines, and said human readable code program is converted by a compiler into machine readable codes of said computer

executable language following predetermined conversion instructions of said conversion rules database.

18. A method of allowing computer programs to be inputted without using advanced programming languages, comprising the steps of:

5 (a) establishing a conversion rule database containing conversion instructions of converting selectable commands to machine readable codes;

(b) providing a selection platform, wherein said selectable commands are listed out for a user to select a set of selected commands according to a desired flow of functions to be performed; and

10 (c) compiling said selected commands into machine readable codes according to said set of conversion instructions.

19. The method, as recited in claim 18, before the step (c), further comprising a sub-step of storing said selected commands inside a processing unit.

15 20. A method of allowing a designed computer program to be customized without using advanced programming languages, comprising the steps of:

(a) establishing a reverse conversion rule database containing reverse conversion instructions of reverse converting machine readable codes of said designed computer program to human understandable codes;

20 (b) establishing a set of conversion rule database containing conversion instructions of converting selectable commands to machine readable codes;

(c) providing an imported code viewing platform, wherein said machine readable codes of said designed computer program are converted to and listed out as said human understandable codes according to said reversion conversion instructions;

25 (d) providing an editing platform, wherein selectable commands are listed out for a user to insert selected commands into said human understandable codes and deleting

sections of said human understandable codes, forming a set of edited codes, according to a desired flow of functions to be performed; and

(e) compiling said edited codes into machine readable codes following said set of conversion rules.